

1 This listing of claims will replace all prior versions, and listings, of claims in the
2 application:

3
4 **Listing of Claims:**

5
6 1. (currently amended) A method comprising:
7 receiving file system data;
8 storing the file system data in a plurality of reserved sectors within a non-
9 volatile memory;

10 compressing the file system data stored within in the plurality of reserved
11 sectors to create a compressed data block; and

12 storing the compressed data block in at least one physical subsector within
13 the non-volatile memory, wherein the physical subsector is associated with at least
14 one virtual sector identifiable through sector allocation information stored in a
15 volatile memory that is operatively accessible by an operating system.

16
17 2. (original) The method as recited in Claim 1, wherein receiving
18 file system data further includes presenting an operating system with a plurality of
19 operatively accessible virtual sectors resulting in a virtual memory capacity that
20 exceeds the actual physical capacity of the non-volatile memory.

21
22 3. (currently amended) The method as recited in Claim 2, wherein
23 storing the compressed data block at least one physical subsector within the non-
24 volatile memory further includes mapping the plurality of virtual sectors to at least
25

1 one physical subsector through a Virtual Sector Table (VST) stored in [[a]] the
2 volatile memory and presenting the operating system with the VST.

3
4 4. (original) The method as recited in Claim 3, wherein mapping
5 the plurality of virtual sectors to at least one physical subsector through the Virtual
6 Sector Table (VST) further includes providing a Sector Allocation Table (SAT)
7 within the volatile memory, the SAT mapping the physical subsectors to the VST.

8
9 5. (original) The method as recited in Claim 4, wherein providing a
10 Sector Allocation Table (SAT) within the volatile memory further includes
11 generating the SAT based at least on a unique group identifier that is stored in each
12 physical subsector associated with storing the compressed data block.

13
14 6. (original) The method as recited in Claim 5, wherein the Sector
15 Allocation Table (SAT) is generated during a device initialization time.

16
17 7. (original) The method as recited in Claim 1, wherein storing the
18 compressed data block in at least one physical subsector within the non-volatile
19 memory further includes associating each physical subsector with a unique group
20 identifier.

1 8. (original) The method as recited in Claim 7, wherein storing the
2 compressed data block in at least one physical subsector within the non-volatile
3 memory further includes writing each physical subsector associated with the
4 compressed data block to the non-volatile memory in an a sequential order, but not
5 necessarily a contiguous order.

6
7 9. (original) The method as recited in Claim 1, wherein storing the
8 compressed data block in at least one physical subsector within the non-volatile
9 memory further includes associating a first physical subsector with at least one
10 virtual sector identifier.

11
12 10. (original) The method as recited in Claim 1, wherein storing the
13 compressed data block in at least one physical subsector within the non-volatile
14 memory further includes, maintaining input/output (I/O) operation status
15 information within the non-volatile memory during on-going I/O operations.

16
17 11. (original) The method as recited in Claim 1, wherein the non-
18 volatile memory further includes a raw sector map configured to identify a
19 plurality of reserved sectors within the non-volatile memory.

20
21 12. (original) The method as recited in Claim 11, wherein the non-
22 volatile memory further includes an input/output (I/O) operation status area that
23 identifies the status of on-going I/O operations with respect to data stored within
24 non-volatile memory.
25

1 13. (original) The method as recited in Claim 11, wherein the non-
2 volatile memory further includes a plurality of contiguously arranged reserved
3 sectors.

4
5 14. (original) The method as recited in Claim 11, wherein the non-
6 volatile memory further includes a plurality of contiguously arranged physical
7 subsectors.

8
9 15. (original) The method as recited in Claim 14, wherein each of the
10 physical subsectors includes a first portion that includes a group identifier.

11
12 16. (original) The method as recited in Claim 14, wherein the group
13 identifier identifies that the physical subsector is unused.

14
15 17. (currently amended) A computer-readable medium having computer-
16 executable instructions for performing steps comprising:

17 receiving file system data;

18 storing the file system data in a plurality of reserved sectors within a non-
19 volatile memory;

20 compressing the file system data stored within in the plurality of reserved
21 sectors to create a compressed data block; and

22 storing the compressed data block in at least one physical subsector within
23 the non-volatile memory, wherein the physical subsector is associated with at least
24 one virtual sector identifiable through sector allocation information stored in a
25 volatile memory that is operatively accessible by an operating system.

1
2 18. (original) The computer-readable medium as recited in Claim 17,
3 wherein receiving file system data further includes presenting an operating system
4 with a plurality of operatively accessible virtual sectors resulting in a virtual
5 memory capacity that exceeds the actual physical capacity of the non-volatile
6 memory.

7
8 19. (currently amended) The computer-readable medium as recited in
9 Claim 18, wherein storing the compressed data block at least one physical
10 subsector within the non-volatile memory further includes mapping the plurality of
11 virtual sectors to at least one physical subsector through a Virtual Sector Table
12 (VST) stored in [[a]] the volatile memory and presenting the operating system with
13 the VST.

14
15 20. (original) The computer-readable medium as recited in Claim 19,
16 wherein mapping the plurality of virtual sectors to at least one physical subsector
17 through the Virtual Sector Table (VST) further includes providing a Sector
18 Allocation Table (SAT) within the volatile memory, the SAT mapping the physical
19 subsectors to the VST.

20
21 21. (original) The computer-readable medium as recited in Claim 20,
22 wherein providing a Sector Allocation Table (SAT) within the volatile memory
23 further includes generating the SAT based at least on a unique group identifier that
24 is stored in each physical subsector associated with storing the compressed data
25 block.

1
2 22. (original) The computer-readable medium as recited in Claim 21,
3 wherein the Sector Allocation Table (SAT) is generated during a device
4 initialization time.

5
6 23. (original) The computer-readable medium as recited in Claim 17,
7 wherein storing the compressed data block in at least one physical subsector within
8 the non-volatile memory further includes associating each physical subsector with
9 a unique group identifier.

10
11 24. (original) The computer-readable medium as recited in Claim 23,
12 wherein storing the compressed data block in at least one physical subsector within
13 the non-volatile memory further includes writing each physical subsector
14 associated with the compressed data block to the non-volatile memory in an a
15 sequential order, but not necessarily a contiguous order.

16
17 25. (original) The computer-readable medium as recited in Claim 17,
18 wherein storing the compressed data block in at least one physical subsector within
19 the non-volatile memory further includes associating a first physical subsector with
20 at least one virtual sector identifier.

21
22 26. (original) The computer-readable medium as recited in Claim 17,
23 wherein storing the compressed data block in at least one physical subsector within
24 the non-volatile memory further includes, maintaining input/output (I/O) operation
25 status information within the non-volatile memory during on-going I/O operations.

1
2 27. (original) The computer-readable medium as recited in Claim 17,
3 wherein the non-volatile memory further includes a raw sector map configured to
4 identify a plurality of reserved sectors within the non-volatile memory.

5
6 28. (original) The computer-readable medium as recited in Claim 27,
7 wherein the non-volatile memory further includes an input/output (I/O) operation
8 status area that identifies the status of on-going I/O operations with respect to data
9 stored within non-volatile memory.

10
11 29. (original) The computer-readable medium as recited in Claim 27,
12 wherein the non-volatile memory further includes a plurality of contiguously
13 arranged reserved sectors.

14
15 30. (original) The computer-readable medium as recited in Claim 27,
16 wherein the non-volatile memory further includes a plurality of contiguously
17 arranged physical subsectors.

18
19 31. (original) The computer-readable medium as recited in Claim 30,
20 wherein each of the physical subsectors includes a first portion that includes a
21 group identifier.

22
23 32. (original) The computer-readable medium as recited in Claim 30,
24 wherein the group identifier identifies that the physical subsector is unused.
25

1 33. (currently amended) An arrangement for use in providing an
2 application access a non-volatile memory, the arrangement comprising:

3 an operating system; and

4 a device driver, wherein the operating system is configured to exchange
5 input/output (I/O) requests with the application and exchange corresponding file
6 system requests with the device driver, and wherein the device driver is configured
7 to store the file system data received from the operating system in a plurality of
8 reserved sectors within the non-volatile memory, compress the file system data
9 stored within in the plurality of reserved sectors to create a compressed data block,
10 and store the compressed data block in at least one physical subsector within the
11 non-volatile memory, wherein the physical subsector is associated with at least one
12 virtual sector identifiable through sector allocation information stored in a volatile
13 memory that is operatively accessible by the operating system.

14
15 34. (original) The arrangement as recited in Claim 33, wherein the
16 device driver is further configured to present the operating system with a plurality
17 of operatively accessible virtual sectors resulting in a virtual memory capacity that
18 exceeds the actual physical capacity of the non-volatile memory.

19
20 35. (currently amended) The arrangement as recited in Claim 34,
21 wherein the device driver is further configured to map the plurality of virtual
22 sectors to at least one physical subsector through a Virtual Sector Table (VST)
23 stored in [[a]] the volatile memory and present the operating system with the VST.
24
25

1 36. (original) The arrangement as recited in Claim 35, wherein the
2 device driver is further configured to store a Sector Allocation Table (SAT) within
3 the volatile memory, the SAT mapping the physical subsectors to the VST.

4
5 37. (original) The arrangement as recited in Claim 36, wherein the
6 device driver is further configured to generate the SAT based at least on a unique
7 group identifier that is stored in each physical subsector associated with storing the
8 compressed data block.

9
10 38. (original) The arrangement as recited in Claim 37, wherein the
11 device driver is further configured to generate the Sector Allocation Table (SAT)
12 during a device initialization time.

13
14 39. (original) The arrangement as recited in Claim 33, wherein the
15 device driver is further configured to associate each physical subsector with a
16 unique group identifier.

17
18 40. (original) The arrangement as recited in Claim 39, wherein the
19 device driver is further configured to write each physical subsector associated with
20 the compressed data block to the non-volatile memory in an a sequential order, but
21 not necessarily a contiguous order.

22
23 41. (original) The arrangement as recited in Claim 33, wherein the
24 device driver is further configured to associate a first physical subsector with at
25 least one virtual sector identifier.

1
2 42. (original) The arrangement as recited in Claim 33, wherein the
3 device driver is further configured to maintain input/output (I/O) operation status
4 information within the non-volatile memory during on-going I/O operations.

5
6 43. (original) The arrangement as recited in Claim 33, further
7 comprising:

8 a processor configured to run the operating system and the device
9 driver; and

10 a non-volatile memory operatively coupled to the processor.

11
12 44. (original) The arrangement as recited in Claim 43, wherein the
13 processor and non-volatile memory are part of a set top box.

14
15 45-46. (canceled)